IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received January 31, 2020, accepted February 23, 2020, date of publication March 2, 2020, date of current version March 13, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2977591

How Much Training Data Is Enough? A Case Study for HTTP Anomaly-Based Intrusion Detection

RAFAEL ESTEPA[1], JESÚS E. DÍAZ-VERDEJO[2], ANTONIO ESTEPA[1], AND GERMAN MADINABEITIA[1]

[1]Department of Telematics Engineering, University of Seville, 41092 Seville, Spain
[2]Department of Signal Theory, Telematics and Communications, CITIC, University of Granada, 18071 Granada, Spain

Corresponding author: Antonio Estepa (aestepa@us.es)

This work was supported in part by the Corporación Tecnológica de Andalucía and the University of Seville through the Projects under Grant CTA 1669/22/2017, Grant PI-1786/22/2018, and Grant PI-1736/22/2017.

ABSTRACT Most anomaly-based intrusion detectors rely on models that learn from training datasets whose quality is crucial in their performance. Albeit the properties of suitable datasets have been formulated, the influence of the dataset size on the performance of the anomaly-based detector has received scarce attention so far. In this work, we investigate the optimal size of a training dataset. This size should be large enough so that training data is representative of normal behavior, but after that point, collecting more data may result in unnecessary waste of time and computational resources, not to mention an increased risk of overtraining. In this spirit, we provide a method to find out when the amount of data collected at the production environment is representative of normal behavior in the context of a detector of HTTP URI attacks based on 1-grammar. Our approach is founded on a set of indicators related to the statistical properties of the data. These indicators are periodically calculated during data collection, producing time series that stabilize when more training data is not expected to translate to better system performance, which indicates that data collection can be stopped. We present a case study with real-life datasets collected at the University of Seville (Spain) and a public dataset from the University of Saskatchewan. The application of our method to these datasets showed that more than 42% of one trace, and almost 20% of another were unnecessarily collected, thereby showing that our proposed method can be an efficient approach for collecting training data at the production environment.

INDEX TERMS Anomaly-based intrusion detection, dataset assessment, training.

I. INTRODUCTION

Anomaly-based Intrusion Detection Systems (AIDS) enable the identification of suspicious behavior that significantly differs from normal activities in a computer system or network [1]. To this end, AIDS model the normal activity of a system adopting diverse approaches (e.g., statistical, knowledge-based, or machine learning techniques) [2]. A prerequisite of AIDS is to train their model with a dataset (*training* dataset) that represents the normal operation of the protected system. Once trained, normal activity profiles are formed and the system performance can be evaluated by rating the events included in a *testing* dataset.

Public benchmark datasets are commonly used to compare different research results [3]. However, in real-life deployments, AIDS need to be trained and validated with datasets that faithfully represent the traffic seen in production. Indeed, inadequate or outdated datasets may lead to false alarms because new behaviors, or changes in the protected system, can be interpreted as anomalies, which is a general issue with AIDS [4]. Therefore, besides their particular models and techniques, the success of anomaly-based detectors strongly depends on the availability of suitable training datasets [5].

The creation of training datasets with real-life properties is not trivial. AIDS in production require to be (re)trained with datasets that at least: (a) are free of attacks (or else these are properly labeled), and (b) represent normal traffic (e.g., up-to-date traffic similar to production). Other desirable properties of a dataset described by Viegas *et al.* [6] include: easily updatable, variant, correct, reproducible (so researchers can compare), and shareable (i.e., with no confidential data).

The associate editor coordinating the review of this manuscript and approving it for publication was Ana Lucila Sandoval Orozco.

Additionally, Sharafaldin *et al.* [7] also pointed to the inclusion of variant protocols and appropriate documentation as two desirable properties of datasets. From the previous requirements, one can infer that extracting suitable datasets from real-life traces is not straightforward and may require a process of sanitization [8], [9] to, at least, identify attacks embedded in the trace (an example of sanitization of HTTP traces can be found in [10]). However, the workload associated with this process grows linearly with the size of the trace. And, although unsupervised sanitization approaches have been suggested (e.g., analysis of entropy [11], or filtering known-attacks with signature-based IDS [12]), manual supervision may be unavoidable in order to discover attacks (e.g., 0-day) unnoticed by fully automated methods [13], [14].

The size of a dataset is a factor that has not received much attention in the scientific literature. One possible reason is that it is commonly assumed to be a given in the experimental outline. A generally accepted idea is that a large volume of data is more representative of normal activity, and as such, it translates to better AIDS performance, which also seems intuitive. Indeed, a tiny dataset may lead to insufficient training and, consequently, poor performance. However, a large dataset may exhibit some drawbacks. First, the data collection may take weeks or even months, which besides increasing the time-to-train the AIDS (and thus, delay the start of operation), can also be associated with higher resource consumption in terms of storage or computational power during data preprocessing or training [15]. This fact might limit applicability in devices with limited processing ability or storage capacity such as those commonly found in industrial control systems, or in the field of IoT (especially with computationally-intensive algorithms [16]). Secondly, the workload associated with the sanitization of a large trace can be prohibitive if done manually, or else, if the sanitization process is fully automated or skipped, the risk of having unnoticed attacks in the resulting dataset increases with the dataset size. Last but not least, large datasets occasionally may lead to the over-training problem in which models are over-adapted to the training set and, as such, AIDS performance deteriorates [17].

In this paper, we investigate the impact of the size of a training dataset on the performance of an anomaly-based intrusion detector. The underlying hypothesis is that there is an optimum size from a cost-benefit perspective, which depends on the detection techniques and model used by the AIDS, as well as the characteristics of the captured traffic [18]. With this in mind, we propose a novel method to find the optimal size of a dataset suited for training AIDS based on 1-grammar models. We use indicators that characterize the learning value of data collected over time. When these indicators stabilize, the amount of data collected is considered optimum for training (i.e., more data would not produce better AIDS performance). A case study applies this methodology to three real-life service traces from our university, and one public dataset from the University of Saskatchewan [19].

The novelty and originality of this work are:

- We study the effect of the dataset size on the performance of detectors of HTTP URI attacks based on 1-grammar models.
- We provide a method to estimate the representativeness of a training dataset with respect to normal behavior, which is applied in a real-life case study.
- We suggest indicators applicable to 1-grammar models that enable the comparison of two evolutionary versions of the same dataset in terms of the training data sufficiency.

The main contribution of this paper is a method to determine when the data collected is representative of normal behavior. This can be useful for reducing the size of existing datasets (e.g., to reduce the risk of overtraining), to reduce the time spent collecting data at the production environment (e.g., to reduce the time needed to put the AIDS in production), or to estimate when (re)training is necessary. Although this work is restricted to AIDS based on 1-grammar, the principles and ideas revealed could be partially reused by the research community to investigate extensions to different models.

The remainder of this paper is as follows. Section II presents related works. Section III introduces the reference AIDS model, definitions and terminology used. Section IV describes the datasets and the resulting dictionaries used in our study. The temporal evolution of these dictionaries is studied in Section V. Our method for on-line data collection is described in Section VI, and Section VII describes the limitations of this work. Finally, Section VIII concludes the paper and outlines future work.

II. RELATED WORKS

As stated earlier, the quality of the datasets used by anomaly-based intrusion detectors has a decisive influence on their performance. In the scientific literature, the performance of different models and techniques is commonly compared using public benchmark datasets whose quality have been subject of criticism by some authors such as Sommer and Paxson [4] or Sharafaldin *et al.* [7]. It is also possible to find some works [18], [20] aimed at defining how to carry out a correct comparison of different AIDS according to the characteristics of the datasets. However, as mentioned earlier, public benchmark datasets, albeit necessary for comparing research results, are not suitable for training models in practice due to the lack of real-life properties similar to those seen in production.

The problem of capturing representative data suitable for training or validating models has been addressed in the past in the research field of machine learning [21], as well as in the anomaly-based intrusion detection research field [22]. The generation of realistic datasets from captured traffic may be a resource-intensive task that some authors have tried to alleviate. In [23], the authors propose techniques for instrumenting network warfare competitions to collect scientifically valid labeled datasets, which otherwise would be resource-intensive. Similarly, Velarde-Alvarado *et al.* [11]

remark the scarcity of suitable datasets for AIDS development and propose a semi-automated process for the sanitization of the traffic captured based on the entropy of embedded traffic flows. This enables the collection of large volumes of data without excessive resource consumption in terms of manual supervision or computational resources. However, as stated earlier, fully automated sanitization methods can never guarantee that the resulting datasets are free of attacks.

Few authors have studied the influence of the training dataset on the performance of the AIDS. In [24], the authors studied the effect of partitioning a dataset to obtain separate pieces for training and evaluation. They found that different data blocks produced different results in AIDS performance, which suggests that the entire dataset exhibited heterogeneous characteristics over time. Maxion and Tan [25], [26] have studied the structure and regularity of captured data and its influence in the performance of an HTTP-attack AIDS based on n-grams. The authors generated artificial datasets of the same size but with increased complexity (according to the relative conditional entropy) obtaining a rate of false positives that increased exponentially with the inverse of the complexity of the data. The authors concluded that training should be adapted to the characteristics of the dataset, including its variability over time, which leads to the consideration of a temporal window in the training dataset. A similar conclusion was drawn by Lee et al. in [27], where the authors analyzed the problems associated with the use of multiple configurations and datasets for evaluating AIDS performance.

The size of the training dataset has received scarce attention in the research literature. Kishimoto *et al.* [17] studied the appropriate size of a learning dataset for anomaly-based intrusion detection based on machine learning. In their work, the authors collected Internet traces from a honeypot and analyzed the effect of trace size on the performance of their classifiers. They found that when the learning dataset was too small (e.g., one day), the rate of false positives was high due to insufficient training. On the other hand, when the size of learning dataset was extremely large (e.g., ten days), overfitting caused the deterioration of performance. Therefore, they experimentally concluded that the appropriate size was five days of capture when using Kyoto2006+ public dataset for validation. This supports our initial hypothesis that there is an optimum size for the training dataset, which depends on the techniques and models used, and the properties of the captured traffic. A similar claim is supported in [28], where the influence of the size of the training data in two classifiers was studied. One of the classifiers compared (Naive Bayes Classifier) was also successfully tested in another work related to anomaly-based intrusion detection [29].

Finally, some works have pointed to the need for re-training the models [4], [5] as a sound solution to the problem of data shift [30]. However, few works actually address the issue of model adaptation to dynamic changes. In [31] the authors propose a batch-based approach that involves manual work to determine the level of performance degradation, which indicates when re-training is necessary. In a more generic context, the authors in [32] have proposed the use of EWMA and Kolmogorov-Smirnoff tests to determine the occurrence of data shift in non-stationary environments. Our contribution can also be applied to find if the dataset used for training is still representative of normal behavior, and therefore, to find whether re-training is necessary or not.

III. REFERENCE AIDS AND TERMINOLOGY

A dataset has to be suited for the specific model and parameters used by the anomaly detector. In this regard, the findings of this work are limited to detectors of anomalous HTTP requests based on probabilistic models (e.g., n-gram [33], or Markov-based models [34]). In particular, the AIDS used in this work is based on 1-grammar since the authors are largely experienced in this technique (see [35]), which is simple enough as to let the reader stay focused on the contribution.

In our reference AIDS, the goal of the training phase is to form a dictionary that will be used afterward to classify Universal Resource Identifiers (URIs) received in HTTP requests as normal or anomalous. As such, the datasets used in this work are a collection of URIs extracted from HTTP trace files.

A. DICTIONARY FORMATION

Let $\mathcal{U} = \{u_i \mid i \in \mathbb{N}\}$ be the set of URIs contained in a training dataset. RFC 3986 [36] defines the structure of a URI, which is basically a text string composed of an optional *protocol*, an optional *host*, a sequence of one or more *path segments*, namely *absolute path* and, optionally, a *query* composed of a sequence of *attributes*, each of them with an optional *value*. This is generally expressed as:
`"http://"host[":"port][abs_path["?"query]]`

A URI can be parsed using a set of standard delimiters (`:/?#[]@!$&′()*+,;=`),[1] obtaining a set of sub-strings or words that are central to our AIDS. For the purpose of anomaly-detection, only those words extracted from the *path*, *attribute* or *value* fields are considered of interest in this work (i.e., we assume that *host* and *port* are invariant throughout the trace).

Let us define the vocabulary learned from a training dataset ($\mathcal{U}$), as the set of words observed after segmenting all the URIs contained in $\mathcal{U}$:

$$\mathcal{W}(\mathcal{U}) = \{w_i \mid 1 \leq i \leq M\} \tag{1}$$

where M is the cardinality of the vocabulary.

Let $\mathcal{O}(\mathcal{U}) = \{o_i \mid 1 \leq i \leq M\}$ be the number of occurrences (i.e., absolute frequency) of the words observed in the training dataset $\mathcal{U}$.

[1] We consider both `gen-delims` and `sub-delims`, as defined in the standard, to be able to parse the queries.

Let us define a dictionary, or equivalently, a 1-grammar[2] as the set of different words (and their absolute frequency) observed after segmenting the URIs contained in a dataset $\mathcal{U}$ as:

$$\mathcal{D}(\mathcal{U}) = \{(w_i, o_i) | w_i \in \mathcal{W}(\mathcal{U}), o_i \in \mathcal{O}(\mathcal{U})\} \tag{2}$$

Given a dictionary, the relative frequency (or empirical probability) of word w_i can be readily obtained as:

$$p_i = \frac{o_i}{O} \tag{3}$$

where O is the overall number of observations:

$$O = \sum_{i=1}^{M} o_i \tag{4}$$

Finally, let $\mathcal{P}(\mathcal{U})$ be the set of relative frequencies of the words observed:

$$\mathcal{P}(\mathcal{U}) = \{p_i | 1 \leq i \leq M\} \tag{5}$$

As an example, consider the following URI:

```
http://traj.us.es/set/index.php?set=200&theme=blue
```

It is possible to segment this URI in 8 strings (`http`, `traj.us.es`, `set`, `index.php`, `set`, `200`, `theme`, `blue`) using standard delimiters. Then, using only strings from the attribute, path and value fields, the extracted dictionary would be:

$$\mathcal{D} = \{(set, 2), (index.php, 1), (theme, 1), (200, 1), (blue, 1)\} \tag{6}$$

The overall number of observations would be $O = 6$, and the relative frequency of the words would be $1/6$ for all but the first one which would be $p_1 = 2/6$.

B. AIDS PERFORMANCE

Figure 1 illustrates a generic scheme for the assessment of the performance of AIDS. This scheme relies on three disjoint datasets:

- Training dataset: it contains URIs that represent normal behavior, and as such, it should be free of attacks. This dataset is used to train the model and is the subject of our study (i.e., $\mathcal{U}$).
- Evaluation dataset (clean): this dataset is also composed of (different) instances from the normal behavior and it should be free of attacks. In this work, the evaluation-clean dataset is similar to the training dataset (indeed, it comes from halving the collected traces).
- Evaluation dataset (attacks): this dataset contains malicious URIs used to evaluate the performance of the detector. In our work, it is composed of 2 200 malicious URIs from a public repository [37] (category ML-driven-Web-Application-Firewall) that can be downloaded from [38].

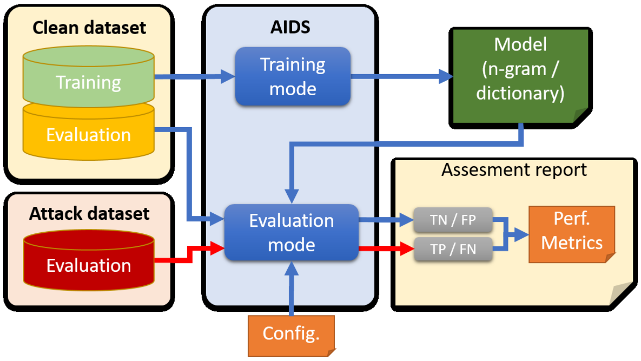


FIGURE 1. AIDS performance evaluation.

After processing the training dataset (training mode in Figure 1), the dictionary is formed and AIDS performance can be evaluated. The AIDS (in evaluation mode) uses the dictionary when assigning an anomaly score to each URI found in the evaluation datasets. Given a URI u_i composed of a set of words $\mathcal{W}_{u_i} = \{w_i | i = 1, \cdots, L\}$, its anomaly score is calculated as follows:

$$AS(u_i) = -\frac{1}{L} \sum_{i=1}^{L} \log(x_i) \tag{7}$$

where

$$x_i = \begin{cases} p_i, & \textit{if } w_i \in \mathcal{W}(\mathcal{U}) \\ p_{oov}, & \textit{if } w_i \notin \mathcal{W}(\mathcal{U}) \end{cases} \tag{8}$$

being p_{oov} a default value assigned to the words not included in the dictionary (i.e., *out of vocabulary* words). In this work, after some tuning, we selected $p_{oov} = p_{min}^3$, where p_{min} is the lowest probability in $\mathcal{P}(\mathcal{U})$. This is an effective solution to deal with the problem of insufficient training [35].

Finally, if the anomaly score exceeds a threshold θ (i.e., $AS(u_i) > \theta$), the URI u_i is classified as anomalous. Otherwise, it is considered normal.

In the evaluation process illustrated in Figure 1, the AIDS classifies registers from the clean datasets as either normal (i.e., True Negative –TN–) or anomalous (i.e., False Positive –FP–), whereas registers from the attack dataset can be classified as either normal (i.e., False Negative –FN–) or anomalous (i.e., True Positive –TP–). These four basic indicators allow one to evaluate AIDS performance through various metrics such as Detection Rate (DR) and False Positive Rate (FPR):

$$DR = \frac{TP}{TP + FN}, \; FPR = \frac{FP}{FP + TP} \tag{9}$$

Other metrics are possible (e.g., accuracy or sensibility) [3], but good performance is always a synonym of very high DR and very low FPR. However since we are going to compare performance in different scenarios, and the classes

[2] An analysis on a per-field basis is also possible by arranging words into field-based dictionaries as in the original SSM technique proposed in [35]. Nevertheless, for the sake of simplicity and clarity, we consider a single state, merging all words in a single dictionary. Experiments carried out using three states did not show differences in the behavior of the proposed method.

normal and anomalous are clearly unbalanced (the attack dataset is several times smaller than the others), we will use the metric *geometric mean* [39] as performance indicator. This metric combines recall and specificity, and hence, it is sensitive to both detection capacity and false positives. Thus, for the remainder of this paper, the AIDS performance metric will be given by:

$$\eta = \sqrt{DR \cdot (1 - FPR)} \tag{10}$$

IV. CHARACTERIZING THE TRAINING DATASETS AND DICTIONARIES OF THIS STUDY

Without loss of generality, for the remainder of this work, we can assume that dictionaries are arrays arranged so words are sorted by their frequency (i.e., words more frequent are first). That is:

$$\mathcal{D}(\mathcal{U}) = \{(w_i, o_i) | 1 \leq i \leq M, o_r \geq o_k \quad \forall r \leq k\} \tag{11}$$

A. STATISTICAL PROPERTIES OF A DICTIONARY

A dictionary $\mathcal{D}(\mathcal{U})$ is statistically characterized by the empirical probability of its words $\mathcal{P}(\mathcal{U})$ (i.e., probability mass function). Since we are assuming that words are sorted by their frequency, a plot of $\mathcal{P}(\mathcal{U})$ should show a monotonically decreasing function such as the one illustrated in Figure 2, where the horizontal axis represents the index of the elements in $\mathcal{P}(\mathcal{U})$ (i.e., word index). Observe that the maximum and minimum empirical probabilities are $p_{\max} = p_1$ and $p_{\min} = p_M$ respectively.

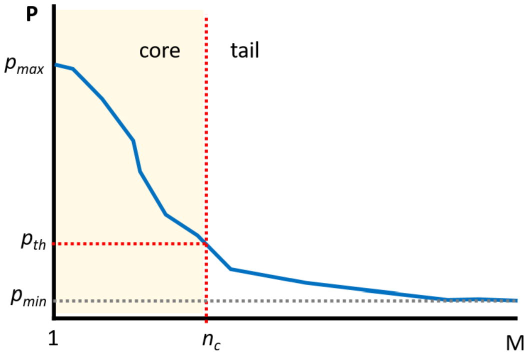


FIGURE 2. **Generic probability mass function of a typical dictionary.**

According to our experience, the probability mass function of URI-based dictionaries is likely to exhibit a tail formed by words rarely observed. If so, the plot of this function can be split into two contiguous regions: core, with the more frequent words, and tail with the less frequent words, by simply defining a lower threshold for the empirical probability of words that belong to the core (see p_{th} in Figure 2). Then, a tail sub-dictionary $\mathcal{T}(\mathcal{U}) \subseteq \mathcal{D}(\mathcal{U})$ can be defined as:

$$\mathcal{T}(\mathcal{U}) = \{(w_i, o_i) | \ (w_i, o_i) \in \mathcal{D}(\mathcal{U}), p_i < p_{th}\} \tag{12}$$

Similarly, a core sub-dictionary $\mathcal{C}(\mathcal{U}) \subseteq \mathcal{D}(\mathcal{U})$ can be defined as:

$$\mathcal{C}(\mathcal{U}) = \{(w_i, o_i) | \ (w_i, o_i) \in \mathcal{D}(\mathcal{U}), p_i \geq p_{th}\} \tag{13}$$

In Figure 2, the number of words that belong to the core is represented by $n_c = |\mathcal{C}(\mathcal{U})|$.

Regarding the value of the threshold p_{th}, it should be lower than the average word frequency, and also should account for the dynamic range of the probability mass function. After some experimentation, we found that the following value provided good results:

$$p_{th} = \frac{1 - (p_{\max} - p_{\min})}{M} = \frac{1 + p_M - p_1}{M} \tag{14}$$

Besides $\mathcal{P}(\mathcal{U})$, a dictionary can be further characterized by the following statistics:

- Average relative frequency of words.

$$R = \frac{M}{O} \tag{15}$$

- Entropy of the dictionary.

$$S = -\sum_{i=1}^{M} p_i \cdot \log_2 p_i \tag{16}$$

- Entropy of the core sub-dictionary.

$$S_{\text{core}} = -\sum_{i=1}^{n_c} p_i \cdot \log_2 p_i \ , \qquad p_i \geq p_{th} \tag{17}$$

- Cumulative empirical probability of core-words:

$$P_{core} = \sum_{i=1}^{n_c} p_i \ , \qquad p_i \geq p_{th} \tag{18}$$

Note that $P_{tail} = 1 - P_{core}$.

Next, we introduce the experimental datasets used in this work and characterize the dictionaries formed after training our reference AIDS with them.

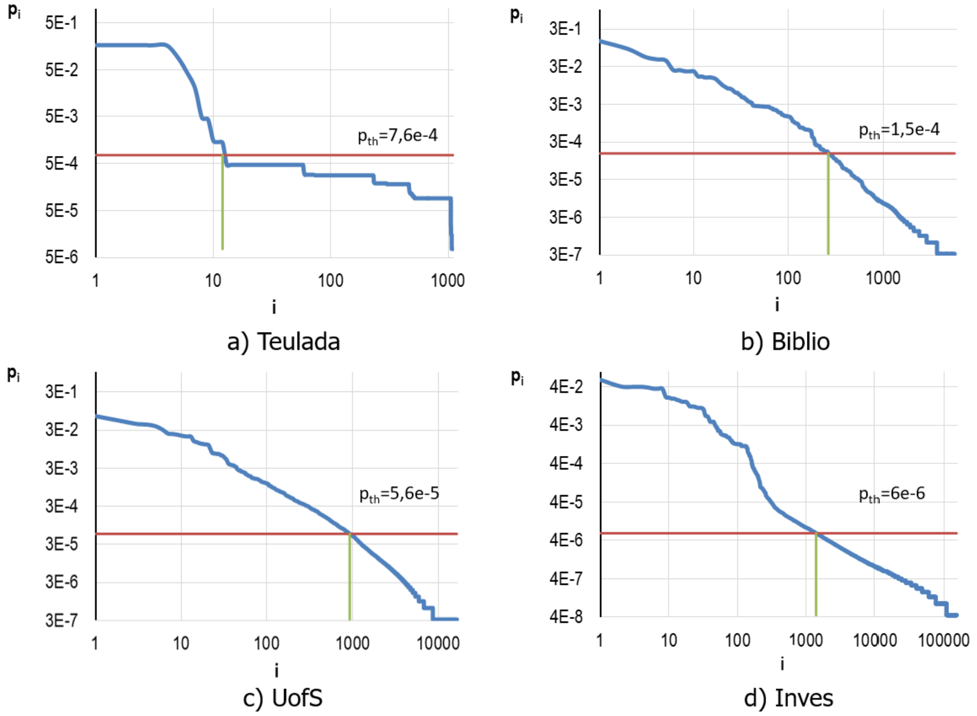
B. TRAINING DATASETS AND BASELINE DICTIONARIES FORMED

In order to perform a comprehensive study, we have used four HTTP traces collected at different experimental testbeds, which has produced datasets with heterogeneous characteristics. The experimental datasets used in this work are:

- *Biblio*: this dataset is formed by the daily traces collected by the web server of the Library of the University of Seville (`http://bib.us.es`). It includes 1 002 000 HTTP requests received from 1/1/2017 to 17/07/2017.
- *Teulada*: this dataset is formed by the traces of a web application server devoted to interwork with a research-oriented IoT sensor network deployed in the city of Seville.
- *UofS*: this is a public dataset from the University of Saskatchewan [19] that includes 2.3 million HTTP requests (method + URI).

TABLE 1. Most relevant statistics for the considered datasets (training partitions).

Datasets	$\lvert\mathcal{U}\rvert(size)$	M	O	S	R	p_{max}	S_{core}	P_{core}	p_{th}	n_c
Teulada	22 K	1 101	131 806	4.5343	0.0084	0.1663	2.6851	0.8321	0.000757	12
UofS	1 185 K	16 577	3 142 799	7.5723	0.0053	0.0699	7.0299	0.9497	0,000056	928
Biblio	501 K	5 706	3 097 204	6.2138	0.0018	0.1444	5.8799	0.9686	0.000150	265
Inves	2 310 K	153 653	22 271 005	7.0718	0.0069	0.0608	6.4447	0.9599	0,000006	1 406



a) Teulada

b) Biblio

c) UofS

d) Inves

FIGURE 3. Probability mass function for the dictionaries formed from: a) Teulada, b) Inves, c) Biblio, d) UofS.

- *Inves*: this dataset is created from the traces of a document-search web service of the University of Seville geared toward research (`http://fama.us.es`). It includes about 4.6 million requests received during May 2018.

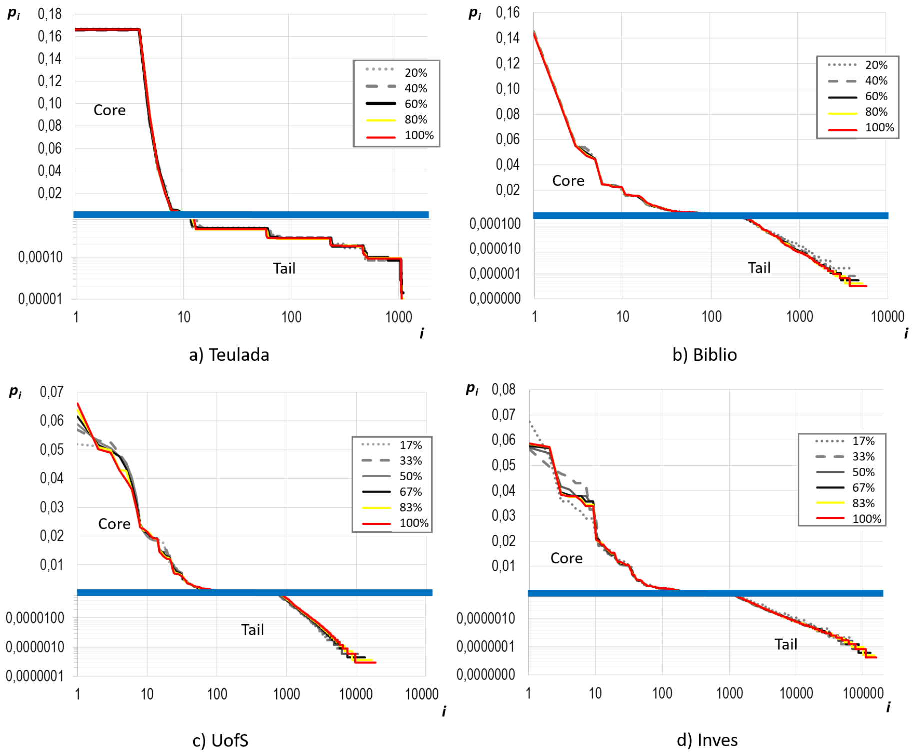
The previous traces have been sanitized to remove existing attacks. Then, the datasets have been halved to create the training and validation (clean) partitions. For the remainder of this paper (but when addressing performance evaluation), we will only refer to the training partition.

Table 1 provides information about the training datasets created on each experimental environment, and some statistics of the dictionary formed with each dataset. These properties show diversity in size, number of words observed, etc. For example, *Teulada* exhibits a reduced vocabulary (e.g., 1 101 different words) while Inves exhibits a large one (153 653 different words). This difference is attributable to the service provided in each case (e.g., Teulada is more similar to a static website whereas Inves provides a search service). This information is complemented with a plot of the mass probability function of the dictionaries formed shown in Figure 3. Note that axes are represented in log scale for clarity, which, albeit improves the visualization of the core, distorts the actual shape.

Results from Table 1 and Figure 3 show that the core sub-dictionary is always composed of a reduced number of words (less than 1% of each vocabulary) that accounts for the bulk of the empirical probability in each vocabulary. As shown in Table 1, core words account for a cumulative probability ranging from 83% (*Teulada*) to 95% (*Biblio*). This suggests that the choice of p_{th}, according to Eq. (39), is reasonable, as tails are commonly expected to represent less than 20% of the distribution. Note also that the entropy of the core is a significant fraction of the entropy of the dataset. On the other hand, the transition between the core and the tail is more abrupt in Teulada and Inves than in Biblio and UofS, which might show that the latter two datasets are more sensitive to the choice of the threshold.

V. TEMPORAL EVOLUTION OF DICTIONARIES

In this section, we study the evolution of the probability mass function of a dictionary with the number of URIs processed.



a) Teulada

b) Biblio

c) UofS

d) Inves

FIGURE 4. Temporal Evolution of the core and the tail in our datasets.

In the datasets, URIs are assumed to be in chronological order. As such, we can regard this study as a temporal evolution. We expect three different behaviors in this temporal evolution that let us classify the dictionaries accordingly as:

1) Stable dictionary: its probability mass function remains mostly steady after a certain number of URIs. This would be the case of websites with a closed set of potential words in their URIs (e.g., static website) when the behavior of users (i.e., requests) is regular over time.
2) Core-stable dictionary: the probability mass function of the core-subdictionary remains mostly steady after processing a certain number of URIs, but it does not stabilize in the tail-subdictionary. It would be the case of websites that include variable parts in their URIs such as timestamps, auto-incremental values, hashes, etc. Although some words are frequently observed (core words), some others (tail words) are scarcely seen (maybe one or two times), having little influence in the anomaly score of a URI.
3) Non-stable dictionary: new URIs might produce significant changes in the probability mass function of the dictionary. As such, neither core nor tail sub-dictionaries stabilize with the number of URIs processed. This is probably the case of dynamic websites with highly changing resources.

Figure 4 shows the temporal evolution of the dictionaries formed with our datasets. For each dataset, we have plotted the mass probability function obtained after processing different percentages of the dataset size. For a clearer view, the tail and core have been represented using two separate scales. The temporal evolution of the tail and the core shown in Figure 4 confirms the types of dictionaries suggested earlier. Teulada remains stable after processing a minimum portion of the dataset (i.e., stable dictionary), Biblio exhibits significant changes in his tail but, after a certain number of URIs, its core remains steady (i.e., core-stable dictionary), and UofS and Inves show unsteadiness in both core and tail sub-dictionaries (i.e., non-stable dictionaries). Regarding the lower threshold probability p_{th} (transition zone in Figure 4), its variations with the number of URIs processed are minimal. Indeed, as demonstrated in Appendix, there is a point after which its variations are negligible.

For the remainder of this section, we investigate how some datasets may hold registers that barely impact the statistical properties of a dictionary. We work under the assumption that in some cases, there should be a minimum-sized dataset that exhibits the same statistical properties as that of a bigger one and, consequently, processing more registers does not pay off from the perspective of AIDS performance. This idea will be used in the next section to stop data collection.

A. STABILITY CONDITIONS OF A DICTIONARY

In this section, we study the conditions that can help us decide when the statistical properties of a dictionary of a certain type become invariant to more data from the same dataset.

1) STABLE DICTIONARY (TYPE 1)

Let $\mathcal{U} = \{u_i | 1 \leq i \leq U\}$ be the set of URIs contained in a dataset in chronological order. Then, we say that a dictionary is stable if there is a value U_s that meets:

$$\mathcal{W}(\mathcal{U}') = \mathcal{W}(\mathcal{U}_s), \quad \forall U' > U_s \tag{19}$$
$$\mathcal{P}(\mathcal{U}') = \mathcal{P}(\mathcal{U}_s), \quad \forall U' > U_s \tag{20}$$

where $\mathcal{U}'$ is the subset composed of the first U' URIs from $\mathcal{U}$, and $\mathcal{U}_s$ is the subset composed of the first U_s URIs from $\mathcal{U}'$.

The first condition –Eq. (19)– can be put in a more tractable form by simply using the cardinality of the vocabularies (i.e., $|\mathcal{W}(\mathcal{U}')| = U_s \ \ \forall U' > U_s$) since $\mathcal{U}_s \subset \mathcal{U}'$ and URIs are processed in chronological order.

The second condition –Eq. (20)–, however, is impossible to meet in a strict sense in practice, as every new URI processed impacts the frequency distribution. Thus, this condition has to be relaxed from equality of distributions to similarity of distributions.

We use the Chi-squared test (χ^2 test) to compare the similarity of two distributions. This test indicates whether there is a significant difference between the expected frequencies and the observed frequencies in one or more categories. In our case, the set of categories is the set of words $\mathcal{W}(\mathcal{U}')$ whose cardinality is M'. Let's assume that $\mathcal{D}' = \mathcal{D}(\mathcal{U}')$ has an unknown probability distribution $\mathcal{P}' = \mathcal{P}(\mathcal{U}')$ and an overall number of observations O', and that $\mathcal{D} = \mathcal{D}(\mathcal{U})$ has a known distribution $\mathcal{P} = \mathcal{P}(\mathcal{U})$. Then, we would like to validate the following hypothesis:

$$H_0 : \mathcal{P} = \mathcal{P}', \tag{21}$$
$$H_1 : \mathcal{P} \neq \mathcal{P}' \tag{22}$$

The χ^2 statistic can be calculated according to the following equation:

$$\chi^2(\mathcal{D}, \mathcal{D}') = \sum_{k=1}^{M'} \frac{(o'_k - p_k \cdot O')^2}{p_k \cdot O'} \tag{23}$$

If $\chi^2(\mathcal{D}, \mathcal{D}')$ is 0, the distribution of the observations in $\mathcal{D}$ and $\mathcal{D}'$ is identical. If not, we can consider that both distributions are similar (i.e., accept the null hypothesis) with a certain statistical significance α if its p_{value} (indicator to support or reject the null hypothesis) is greater than α (i.e., $p_{value}(\mathcal{D}, \mathcal{D}') = Prob(\chi^2_{(M'-1)} > \chi^2(\mathcal{D}, \mathcal{D}')) > \alpha$)).[3]

Finally, notice that in order to have a reliable application of the Chi-squared test, the following has to be met: (a) the overall number of observations has to be large, (b) the frequency of each word should be greater than a lower threshold (typically 2). In our case, the first condition is met in all datasets, and the second condition has been applied by excluding words whose frequency is less than that lower threshold from both $\mathcal{D}$ and $\mathcal{D}'$.

Therefore, the requirement for a dictionary to be considered stable with a statistical significance of α, is that there is a value U_s that meets the following conditions:

$$|\mathcal{W}(\mathcal{U}')| = |\mathcal{W}(\mathcal{U}_s)|, \quad \forall U' > U_s \tag{24}$$
$$p_{value}(\mathcal{D}, \mathcal{D}') > \alpha, \quad \forall U' > U_s \tag{25}$$

2) CORE-STABLE DICTIONARY (TYPE 2)

In this case, both the mass probability function and the number of words in the core-subdictionary stabilize after some point, but the number of new words (in the tail) is continuously growing. Therefore, this type of dictionaries can be characterized by:

$$|\mathcal{W}(\mathcal{U}')| \geq |\mathcal{W}(\mathcal{U}_s)|, \quad \forall U' > U_s \tag{26}$$
$$\mathcal{C}(\mathcal{U}') = \mathcal{C}(\mathcal{U}_s), \quad \forall U' > U_s \tag{27}$$

In this case, stability conditions can be set based on the stability of the core-subdictionary $\mathcal{C}(\mathcal{U})$:

$$|\mathcal{C}(\mathcal{U}')| = |\mathcal{C}(\mathcal{U}_s)|, \quad \forall U' > U_s \tag{28}$$
$$p_{value}(\mathcal{C}(\mathcal{U}'), \mathcal{C}(\mathcal{U}_s)) > \alpha, \quad \forall U' > U_s \tag{29}$$

As shown in Appendix, in type 2 dictionaries, p_{th} tends to stabilize after a certain point, and so does the number of core-words and their cumulative probability. Nevertheless, $p_{value}(\mathcal{C}(\mathcal{U}'), \mathcal{C}(\mathcal{U}_s))$ is particularly sensitive to fluctuations in the core-tail delimitation, which depends on p_{th}. Therefore, it would be desirable to replace Eq. (29) with an alternative condition that let us compare the similarity of the distributions and has better tractability. We believe that entropy, although is a softer condition, can be used to this end.

Lemma 1: Given a core-stable dataset $\mathcal{U}$, of size U, there is a minimal subset, of size $U_s < U$, whose entropy would be equal to the entropy of the full dataset if U was large enough.

Proof: see Appendix. ■

Then, for the remainder of this work, we consider that a dictionary is core-stable if there is a value U_s that meets Eq. (28) and:

$$S(\mathcal{U}') = S(\mathcal{U}_s) \pm \sigma, \quad \forall U' > U_s \tag{30}$$

where σ is a constant that accounts for minimal differences which tends to 0 with the dataset size.

Obviously, those dictionaries that can not be classified as type-1 or type-2, will be considered non-stable (type 3)[4] for which no stability conditions can be set.

B. APPLICATION OF THE STABILITY CONDITIONS TO FIND DISPENSABLE DATA IN OUR DATASETS

In this section, we seek the previous stability conditions in our experimental datasets in order to find the minimum value of U', namely U_s, so that the statistical properties of the resulting dictionary were similar to that of the dictionary formed with the full dataset. In this process, we first divide the original datasets in data chunks of ΔU URIs. Then, we look

[3] Typical accepted values for α are: 0.05, 0.01 and 0.001.

[4] A given training dataset can be classified as type 3 even if its associated source could correspond to type 1 or type 2 due to insufficiency of the acquired training dataset. In any case, the conclusion is that additional observations are needed.

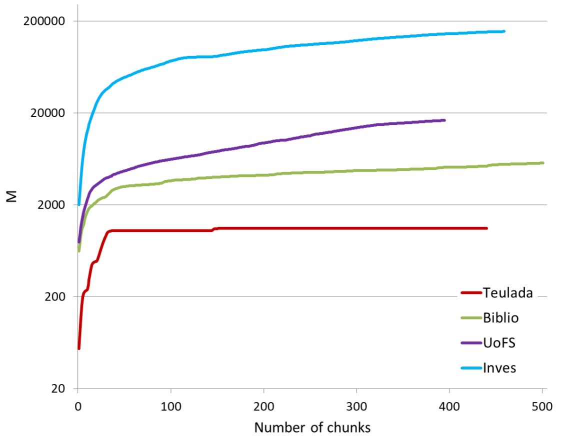


FIGURE 5. Evolution of the number of words (*M*) for different dataset sizes.

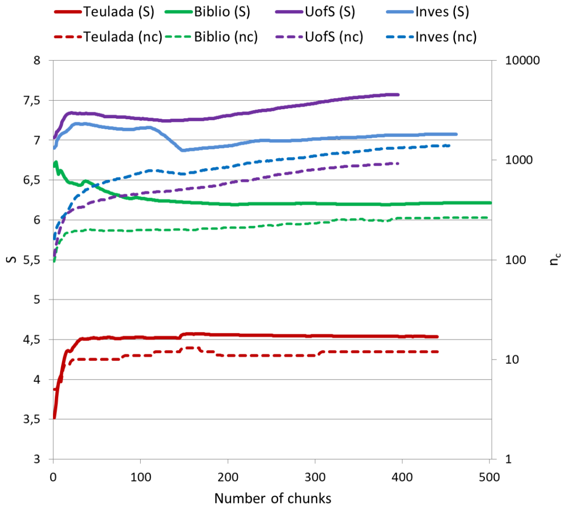


FIGURE 6. Evolution of the Entropy (*S*) and word in the Core (n_c) for different dataset sizes.

for the minimum number of pieces that meets the stability conditions.

The first condition for type 1 dictionaries is related to the number of words contained in the dataset–Eq. (24)–. Figure 5 shows the evolution of this indicator (M) with the number of chunks processed for each dataset under study. The size of the chunk on each case is a divisor of the dataset size.

The results in Figure 5 show that only Teulada meets this condition. The other requirement for a dataset to be considered type 1 was similarity of distribution –Eq. (25)–. Therefore, we want to find the minimum value of U', namely U_s so both dictionaries $\mathcal{D}(\mathcal{U})$ and $\mathcal{D}(\mathcal{U}_s)$ are similar with a statistical significance of α.

Figure 6 shows the evolution of the entropy and the number of core-words (n_c) in the datasets. It can be noticed that, besides Teulada, only Biblio exhibits a stable behavior. Thus, according to Lemma 1 and Eqs. (25), (28) and (29), Teulada is definitively a dataset that produces a stable (type-1) dictionary, whereas Biblio can be classified as core-stable (type-2).

Algorithm 1 U_s Search Algorithm for Stable (Type=1) and Core-Stable (Type=2) Dictionaries

```
Input: $\mathcal{U}$, $U$, $\alpha$, $\Delta U$, $\sigma$, $Type$
Output: $U_s$
function: $D(n)$
    $d = \{u_i \in \mathcal{U} | i \leq n\}$
    return $d$
end function
$\mathcal{D} \leftarrow D(U)$
$U' = U - \Delta U$
$\mathcal{D}' \leftarrow D(U')$
if $Type = 1$ then
    while $((\alpha \leq p_{value}(\mathcal{D}, \mathcal{D}'))$ & $(|\mathcal{W}(\mathcal{U})| = |\mathcal{W}(\mathcal{U}')|)$ & $(U' > \Delta U))$ do
        $U' \leftarrow U' - \Delta U$
        $\mathcal{D}' \leftarrow D(U')$
    end while
    if $\alpha > p_{value}(\mathcal{D}, \mathcal{D}')$ then
        $U_s \leftarrow U' + \Delta U$
    else
        $U_s \leftarrow U'$
    end if
else
    while $((\sigma \leq |S(\mathcal{D}) - S(\mathcal{D}')|)$ & $(|\mathcal{W}(\mathcal{C}(\mathcal{U}))| = |\mathcal{W}(\mathcal{C}(\mathcal{U}'))|)$ & $(U' > \Delta U))$ do
        $U' \leftarrow U' - \Delta U$
        $\mathcal{D}' \leftarrow D(U')$
    end while
    if $\sigma > |S(\mathcal{D}) - S(\mathcal{D}')|$ then
        $U_s \leftarrow U' + \Delta U$
    else
        $U_s \leftarrow U'$
    end if
end if
return $U_s$
```

As such, we want to find the minimal subset U_s so the difference of the entropy in both dictionaries $\mathcal{D}(\mathcal{U})$ and $\mathcal{D}(\mathcal{U}_s)$ is lower than σ.

Algorithm 1 shows a pseudocode that finds the value of U_s in datasets that produce stable or core-stable dictionaries. It takes as input the initial dataset considered, $\mathcal{U}$, its size, U, and type, the α value considered for similarity's statistical significance, the σ value considered in Eq. 30, and the dataset size reduction step Δ. It first builds the dictionary $\mathcal{D} = D(U)$ using the full dataset. Then, it builds a second dictionary $\mathcal{D}' = D(U')$ that excludes the last ΔU URIs. The similarity condition (p_{value} or Entropy difference) is then examined and, if met, the size is reduced by another ΔU URIs, and $\mathcal{D}'$ is rebuilt. Then, the similarity condition is examined again. This process continues until both are not similar, or the dataset size cannot be further reduced. The algorithm returns U_s. Note that U_s, will be lower than U only when it is applied to the correct dataset type. Thus, if applied to Inves and UofS,

TABLE 2. Dataset size reduction results.

Datasets	**Step(ΔU)**	U	U_s	ΔM	*Reduction*	Type
Teulada	50	22 000	4 100	0%	81.13%	1
UofS	3 000	1 185 000	1 185 000	0%	0%	3
Biblio	1 000	501 000	352 000	-1.79%	29.7%	2
Inves	5 000	2 310 000	2 310 000	0%	0%	3

TABLE 3. Performance of the AIDS after training datasize reduction.

Dataset	$\eta(\mathcal{D}(U))$	$\eta(\mathcal{D}(U_s))$	$\eta_{\max}$	$\eta_{\min}$	$\Delta\eta$
Teulada	1.0000	1.0000	1.0000	0.9686	0.00%
UofS	0.9501	0.9501	0.9501	0.8961	0.00%
Biblio	0.9384	0.9407	0.9407	0.9367	0.24%
Inves	0.9501	0.9501	0.9819	0.9650	0.00%

the value of U_s returned should be similar to U. Also note that datasets that produce type 1 dictionaries, like Teulada, also meet the type 2 conditions. In this case, the most restrictive option (Type=1) will be applied.

The results of applying Algorithm 1 to our experimental datasets are shown in Table 2. The parameters used were: $\alpha = 0.001$, $\sigma = 0.0001$ and the step reduction size ΔU (shown in column 2) has been set proportional to the size of the dataset to avoid unnecessary computation. Column 5 in Table 2 (ΔM), stands for the percentage of missing words in $\mathcal{D}(U_s)$ with respect to $\mathcal{D}(U)$. The last two columns (*Reduction* and *Type*) show the percentage of URIs from the original dataset excluded in the smallest dataset and the type of dictionary formed, respectively.

Note that only Teulada is stable and Biblio is core-stable, so dataset size reduction ($U_s < U$) can only be obtained in these cases (81.13% for Teulada and 29.7% for Biblio). Observe also that the dictionary formed with the shortest dataset from Biblio contains fewer words (a 1.79% less) than that formed with the full dataset. Finally, as expected, UofS and Inves datasets do not achieve any reduction of their size as in type-3 datasets stability conditions do not hold. This simply suggests that in these two datasets, the amount of data collected may not be sufficient to be representative of normal behavior.

Next, we examine the performance of our AIDS trained with the smallest dataset of size U_s.

C. PERFORMANCE OF OUR AIDS TRAINED WITH THE SMALLEST DATASET

In this section, we follow the procedure described in Section III-B to assess AIDS performance when trained with both dictionaries, i.e., $\mathcal{D}(\mathcal{U}_s)$ and $\mathcal{D}(\mathcal{U})$. According to our hypothesis, performance degradation should pass unnoticed in stable dictionaries such as Teulada. For core-stable dictionaries, performance degradation mainly depends on the Anomaly Score calculation method. In our case, the impact is expected to be proportional to the probability that a word from a new URI belongs to the tail sub-dictionary. Therefore, a very limited impact it is expected.

On each experiment, we have set the decision threshold θ so that η is maximized (in order to be able to compare the best case on each dataset). We have also gauged η every time that we have decreased the training dataset by ΔU in our previous algorithm.

Table 3 shows the performance obtained with $\mathcal{D}(\mathcal{U})$ (i.e., $\eta(\mathcal{D}(\mathcal{U}))$), and also with $\mathcal{D}(\mathcal{U}_s)$. For comparative purposes, Table 3 also shows the maximum and minimum values of η achieved over the process. The last column ($\Delta\eta$) indicates the difference in performance obtained as a result of using $\mathcal{D}(\mathcal{U}_s)$ instead of $\mathcal{D}(\mathcal{U})$. Results suggest that the AIDS performance is not impacted despite the significant size reduction in Teulada, which confirms our initial hypothesis. In the case of Biblio, the size reduction using the conditions of core-stable dictionaries has improved the AIDS performance, which suggests that the Biblio dataset exhibits an over-training problem. This can be inferred from the fact that there is a peak of $\eta_{max} = 0.9407$ with a dataset size smaller than U. After this point, performance decreases to $\eta = 0.9384$ with more URIs.

Experimental results have confirmed our hypothesis. However, despite its theoretical interest, reducing the size of a dataset that has already been collected has a limited interest in practice since it requires to have the full trace (i.e., dataset) plus the algorithm perform several training stages for comparison. Thus, although the risk of overtraining may be reduced, no resource savings is obtained (e.g., time, sanitization, computation).

Next, we apply our previous findings to develop a method for real-time collection of traces.

VI. SCHEME FOR ON-LINE DATA COLLECTION

Data collection for anomaly detection has traditionally be founded on the belief that larger data volumes produce better AIDS performance due to better representativeness of normal system behavior. Nevertheless, in the previous section, we showed that this is not always the case, especially in datasets that produce stable dictionaries. In this section, we apply our findings to design a procedure that determines the sufficiency of a dataset during its acquisition.

A. COLLECTION SCHEME

In Section V-A we showed how, according to the type of dataset, some properties of a dictionary tend to stabilize after processing a certain number of URIs. Based on this idea, in this section we suggest a simple collection scheme based on the observation of a set of indicators. This idea is illustrated in Figure 7. We start the dataset collection with a minimum chunk of registers, $\mathcal{U}(0)$, of size n_0, that is,

$$\mathcal{U}(0) = \{u_i | 1 \leq i \leq n_0\} \tag{31}$$

Then, when a step of ΔU additional registers are collected, we (re)evaluate the dataset,

$$\mathcal{U}(n) = \{u_i | 1 \leq i \leq n_o + n\Delta U\} \tag{32}$$

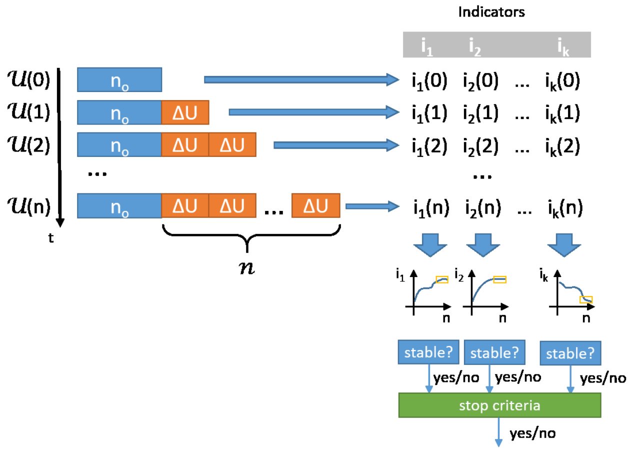


FIGURE 7. Algorithm for on-line capture.

TABLE 4. Stability conditions considered.

	M	p_{value}	S	n_c
type 1: stable dictionary	yes	yes (full dictionary)	yes	yes
type 2: core-stale subdictionary	no	no	yes (core)	yes
type 3: otherwise	no	no	no	no

where n stands for the step index. During each evaluation, a set of indicators are calculated. After a number of evaluations, these indicators can be interpreted as a set of discrete temporal series used to determine when the collection of data can be interrupted according to a stop criteria based on the stabilization of the temporal series.

Next, we discuss the indicators proposed and the possible stability conditions that flag the end of the collection.

B. INDICATORS AND STABILITY

As studied in Section V-A, some properties of a dictionary tend to stabilize with the number of URIs processed according to its type (see Table 4 for a summary).

Based on these properties, and in order to evaluate the dataset $\mathcal{U}(n)$, the following indicators will be calculated every time that ΔU new URIs are collected.

1) INDICATORS DEFINITION

- i_1, vocabulary size: i_1 is an integer that stands for the number of different words observed since the beginning of data collection. Therefore, at step n, the number of elements in the vocabulary is calculated as:

$$i_1(n) = |\mathcal{W}(\mathcal{U}(n))| = M(n) \tag{33}$$

- i_2, index of the most recent smallest dictionary with similar distribution: remember that p_{value} can be used to check if two dictionaries $\mathcal{D}$ and $\mathcal{D}'$ have a similar distribution with statistical significance α.

  In this case, at a given step n, we would like to find out the smallest value of the index $n_{\min}$ for which the dictionary $\mathcal{D}(\mathcal{U}(n_{\min}))$ bears statistical similarity with the last dictionary collected $\mathcal{D}(\mathcal{U}(n))$. This can be achieved with Algorithm 2:

Algorithm 2 Minimum Slot Search Algorithm

```
Input: $\mathcal{U}(n),\ n_o,\ n,\ \Delta U$
Output: $n_{\min}$
$\mathcal{D}' \leftarrow \mathcal{D}(\mathcal{U}(n))$ // last collected
$i \leftarrow 0$
while ($i \leq n$ ) do
    $\mathcal{D} \leftarrow \mathcal{D}(\mathcal{U}(i)) = \mathcal{D}(\{w_j \in \mathcal{U}(n) | 1 \leq j \leq n_0 + i\Delta U\})$
    if $\alpha > p_{value}(\mathcal{D}, \mathcal{D}')$ then
        $n_{\min} \leftarrow i$
        break;
    else
        $i \leftarrow i + 1$
    end if
end while
return $n_{\min}$
```

  Note that dictionaries $\mathcal{D}(\mathcal{U}(i))$, $n_{\min} < i < n$, also bear statistical similarity with $\mathcal{D}(\mathcal{U}(n))$. Since we would like to have growing values for this second indicator, we will define it as:

$$i_2(n) = \max(i_2(n-1), Algorithm2(\mathcal{U}(n), n_0, n, \Delta U)) \tag{34}$$

  This indicator is expected to remain constant in datasets that produce type 1 dictionaries.

- i_3, entropy: this indicator simply returns the entropy of the considered dictionary. Therefore, given a step, n, it is calculated as:

$$i_3(n) = S\left(\mathcal{D}(\mathcal{U}(n))\right) \tag{35}$$

  where S is calculated as expressed in Eq. (16). The entropy tends to stabilize in datasets that produce dictionaries of type 1 and 2.

- i_4, the number of words that belong to the core: given $\mathcal{U}(n)$, the core sub-dictionary $\mathcal{C}(\mathcal{U}(n))$ is obtained. Then $i_4(n)$ is calculated as:

$$i_4(n) = |\mathcal{C}(\mathcal{U}(n))| = n_c(n) \tag{36}$$

  Recall that this indicator tends to stabilize in datasets that produce dictionaries of type 1 and 2.

2) STABILITY CRITERIA

A central aspect of our method is the actual criteria used for considering when each of the previous indicators has stabilized. For indicators i_1, i_2, i_3, and i_4, our stability criterion will be based on the Exponentially Weighted Moving Average (EWMA) [40]. This method obtains the average value of a sequence of sample values considering recent values with an exponential weight and a decay factor, λ. EWMA has been proved to be a good estimator to forecast future values of a

Algorithm 3 Dataset Collection Algorithm

```
Input: $\Delta U$, $n_0$, $L$, $U_{\max}$
Output: $\mathcal{U}$
$n \leftarrow 0$
$\mathcal{U} \leftarrow$collect($n_0$)
while (($n \cdot \Delta U$) $<$ $U_{\max} - n_0$) do
    $n \leftarrow n + 1$
    $\mathcal{U} \leftarrow \mathcal{U} \cup$ collect($\Delta U$)
    calculate $i_1(n)$, $i_2(n)$, $i_3(n)$, $i_4(n)$
    calculate stability for all indicators
    if $n \geq L$ then
        calculate $\delta(n)$ for all indicators
        check stability condition for all indicators
        if ($i_1$ & $i_2$ are stable) | ($i_3$ & $i_4$ are stable) then
            break
        end if
    end if
end while
return $\mathcal{U}$
```

variable [41], [42]. The value of EWMA at the instant n for the indicator i_x is given by:

$$E(n) = \lambda \cdot i_x(n) + (1 - \lambda) \cdot E(n-1) \tag{37}$$

To determine the stabilization point of the series $E(n)$, we use a temporal window of L samples, over which we will calculate the variation coefficient, δ, as the ratio of the sample standard deviation and the mean:

$$\delta(n) = \frac{\sigma}{\mu} = \frac{\sqrt{1/L \sum_{i=1}^{L} (E(n-i) - \bar{E})^2}}{\bar{E}}, \quad \forall n > L \tag{38}$$

where $\bar{E}$ stands for the mean value of $E(n)$ in the sliding window of L samples, i.e., $\bar{E} = 1/L \sum_{i=1}^{L} E(i)$.

Then, indicators $i_1(n)$, $i_2(n)$, $i_3(n)$, and $i_4(n)$ are considered stable when their coefficient $\delta(n)$ is lower than a certain threshold (ϕ) for a number of consecutive times (l_{min}). Values for ϕ and l_{min} should be adjusted on a per case basis. Typical values used in our experiment are $\phi = 10^{-4}$ and $l_{min} = 15$, although these parameters could be individually arranged for each indicator.

3) STOP CRITERIA

The stop criteria follows the stability conditions shown in Table 4. As such, if i_1 and i_2 stabilize, the dictionary is considered type 1 and therefore, the collection can stop. However, if either i_1 or i_2 do not stabilize but i_3 and i_4 do, the dictionary is considered type 2 and the collection stops. Otherwise, the collection stops when the capture reaches an administrative value U_{max}.[5] In Algorithm 3 we show a high-level pseudocode of the collection algorithm proposed.

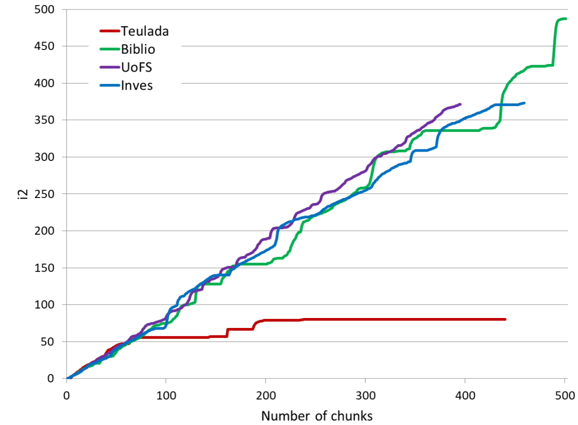


FIGURE 8. Evolution of the indicator i_2 for the different datasets.

TABLE 5. Stabilization point for each indicator and breakpoint for collection.

	i_1	i_2	i_3	i_4	$U(Stop)$
Teulada	≥ 37 (1 850)	≥ 251 (12 550)	≥ 306 (15 300)	≥ 201 (10 050)	12 550
UofS	-	-	-	-	1 185 000
Biblio	-	-	≥ 341	≥ 404	404 000
	-	-	(341 000)	(404 000)	
Inves	-	-	-	-	2 310 000

C. APPLICATION TO OUR EXPERIMENTAL DATASETS

We have applied the proposed collection method to our datasets by re-collecting their URIs in chronological order. This will let us study the evolution of the indicators, find the size that would have been captured with our proposal, and most importantly, verify that the size reduction in Biblio and Teulada has a limited impact on AIDS performance.

The evolution of indicators i_1, i_3 and i_4 has been already shown in Figures 5 and 6. Figure 8 shows the evolution of i_2 with the dataset size for the four training datasets under study. It can be observed that, as expected, i_2 only stabilizes in the case of Teulada (stable-dictionary, type 1). The stabilization point for these indicators is shown in Table 5, where column U stands for the number of URIs processed until the stop criteria is met. The other columns from Table 5 show both the number of steps in which the algorithm has met the stop criteria, and the number of URI collected at such point (in brackets). Results show that our on-line method accurately 'detects' the different dictionary types, and saves a significant number of URIs in the collection phase for the dictionaries Teulada and Biblio (stable and core-stable dictionaries respectively). For type-3 dictionaries (Inves and UofS) our method only stops after reaching the maximum administrative value (U_{max}).

In order to evaluate the influence of our on-line collection method in AIDS performance, we have followed a procedure similar to that of subsection V-C. Table 6 shows the results, where column $\eta(U)$ shows the performance obtained when the AIDS is trained with the datasets determined by our algorithm; column $\eta(U_{max})$ shows the AIDS performance

[5] In a softer alternative approach, the collection could be stopped when variations in i_3 and i_4 were considered very small.

TABLE 6. Size reduction and performance comparison.

	$Reduction$	$\eta(U)$	$\eta(U_{max})$	$\Delta\eta$
Teulada	42.95%	100.00%	100.00%	0.00 %
UofS	0.00%	95.01%	95.01%	0.00 %
Biblio	19.36%	94.07%	93.84%	0.24 %
Inves	0.00%	98.19%	98.19%	0.00 %

obtained with the full training dataset, and column $\Delta\eta$ stands for the difference in performance (i.e., $\eta(U) - \eta(U_{max})$). The dataset size reduction obtained with Algorithm 3 is also shown in column *Reduction* ($= 1 - (U/U_{\max})$). It can be observed that, despite a size reduction of 42.95%, the AIDS performance is not impacted in Teulada.[6] In the case of Biblio, we obtain a size reduction of 19.36%, but the AIDS performance is slightly better than obtained with the full dataset. This suggests that the algorithm also works with overtrained datasets. In the case of Inves and UofS, our algorithm produced no size reduction, and thereby unaffected AIDS performance.

D. COMPUTATIONAL EFFORT

In order to apply the proposed algorithm in real-time, the time devoted to processing a new chunk with ΔU URIs should be less than its collection time. For each URI processed, the computational effort includes vocabulary extraction and dictionary update (including searching in a dictionary with M words). After that, indicators i_1 to i_4 should be calculated and stability and stop criteria computed. This exhibits a complexity of $O(\Delta U \cdot M \log M)$. In Figure 9 we show the execution time of the on-line collection algorithm implemented in Python and ran in a computer with a Xeon(R) E5645@2.40GHz processor. As expected, the execution time grows with the chunk size (ΔU) and the vocabulary size (M). Note that for 100 000 URIs, the vocabulary size in UofS and Biblio is about 3 800 words, while for Inves is about 27 700 words. This results in a different slope for the Inves dataset in Figure 9. In the worst case, Inves, we achieved a throughput of 4 500 URIs by second, which suffices to be applicable to most real-life scenarios.

VII. LIMITATIONS OF THIS WORK

This work is only a first step into the study of the impact of the volume of training data on AIDS performance. Besides the fact of studying four specific datasets, several factors limit the generalization of the results presented in this work. The main ones are:

- We implicitly assume that, after the stabilization of some indicators, the statistical properties of future URIs will not change. This ergodic behavior could be a hard assumption in many real scenarios. Therefore, one should periodically verify if this assumption holds and, if not, initiate a new collection period for re-training the AIDS.
- The parameters used have a decisive impact on the results. For example, $U_{\max}$, the parameters used for stability δ and L, the number of URIs to be initially collected until p_{th} is considered stable, the step size ΔU, etc. Researchers are advised to investigate and perform fine tuning of these parameters on a per case basis.
- The dictionary-based model (or 1-grammar) is only one possible method that AIDS can employ to detect anomaly in HTTP attacks [2]. Thus, this work is limited by this and it is only applicable to similar models.

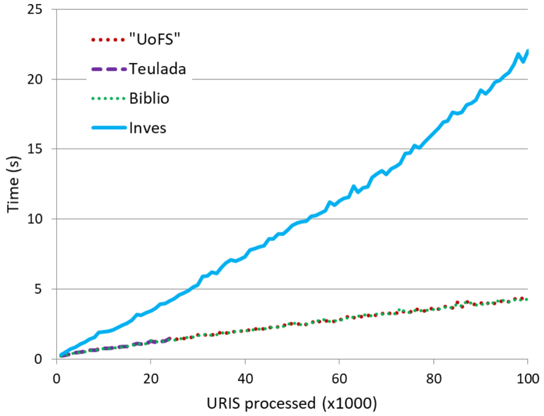


FIGURE 9. Computation time for different chunk sizes.

These limitations pave the way for further research, as discussed next.

VIII. CONCLUSIONS AND FURTHER WORKS

In this work, we have investigated the size of the training dataset and its effect on AIDS performance in the context of HTTP URI-attack detection. We have proposed a set of stability conditions to determine when data captured at the production environment is representative of normal behavior and, as such, is suited for training an AIDS based on 1-grammar. Using these conditions, we have proposed a method to determine in real-time when the data collection can be stopped. This method has been applied to four real-life datasets concluding that, in some cases, smaller datasets would have produced the same AIDS performance. Thus, depending on the properties of the data being captured, it may be possible to perform a more efficient data collection in anomaly detectors similar to the one used in our study. Collecting less data will allow one to reduce the cost of collection and sanitization of real-life traces, and reduce the risk of overtraining. The former includes reducing the data collection time span which might speed up the time to (re)training and facilitate an earlier deployment of the AIDS.

A number of issues can be appointed for further work. A sensitivity study of the parameters used would be interesting to provide finer guidance to the reader in the application

[6] Note that datasets that produce stable dictionaries also meet the stop criteria for core-stable dictionaries. To sort this out, we could add extra time (an extra transient period) to allow for the stabilization of indicators i_1 and i_2. In our experiments, however, stabilization of i_1 and i_2 happened earlier than i_3 and i_4.

of the proposed collection method. A finer study of type-3 dictionaries might produce new stability conditions with affordable performance degradation. We would also like to explore the use of our method for deciding when re-training is necessary by comparing the properties of the current dictionary with that of the dictionary that would be obtained with current traffic. Finally, the extension of our method to other models such as n-grammars ($n > 1$) should also be investigated. In particular, the use of sequences of words rather than word frequencies would increase the possibility of applying other information-theoretical concepts like the asymptotic equipartition property.

APPENDIX

ROOF OF LEMMA

Lemma 1 states that:

Given a core-stable dataset $\mathcal{U}$, of size U, there is a minimal subset, of size $U_s < U$, whose entropy would be equal to the entropy of the full dataset if U was large enough.

Proof: Remember that core-stable dictionaries include not only n_c core-words, but also tail-words (i.e., rarely observed words, maybe one or two times) which commonly have little impact in the anomaly score of a URI. We can assume that tail-words have a similar common frequency (a small number, A). As such, the tail-words' empirical probability can be assumed to be $p_i = A/O$, $\forall w_i \in \mathcal{T}(\mathcal{U})$ (where O was the overall number of observations in the dictionary $\mathcal{D}(\mathcal{U})$).

We are interested first in confirming that p_{th} stabilizes for larges values of U. To this end, we can rewrite p_{th} (see Eq. (39)) taking into account that the cardinality of the dictionary can be also expressed as $M = n_c + O \cdot P_{tail}/A$, (where n_c was the number of core-words in the dictionary). Then:

$$p_{th} = \frac{1-(p_{\max}-p_{\min})}{M} = \frac{1-p_{max}+A/O}{n_c+O\cdot P_{tail}/A} \qquad (39)$$

Note that O grows at least linearly with U (i.e., in each URI there will be at least one word). Then, for very large values of U ($U \gg 1$), $A/O \rightarrow 0$, and, since core-words probabilities remain stable, we can assume that p_{max} and n_c remain stable. Then we can state that for a large value of U, p_{th} tend to stabilize, or equivalently:

$$\begin{aligned} p_{th}&(U+\Delta U) - p_{th}(U) \\ &= (1-p_{max})\cdot(\frac{1}{M+\Delta M} - \frac{1}{M}) \\ &= \frac{1-p_{max}}{n_c + O\cdot P_{tail}/A\cdot(\frac{n_c O}{\Delta O}+1)} \xrightarrow[U\gg 1]{} 0 \end{aligned}$$

where ΔO is the increment in words observations associated to the new ΔU URIs ($\Delta U \ll U$).

Since p_{th} can be considered constant for larges values of U, and the probabilities p_i of the words in the core remains the same according to Eq. (27), n_c does not change either. This lead us directly to:

1) $S_{core}(\mathcal{U}) = S_{core}(\mathcal{U}_s)$ (due to condition in (27))
2) $P_{tail}(\mathcal{U}) = P_{tail}(\mathcal{U}_s)$, (as $P_{tail} = 1 - P_{core}$).

Due to 1) any difference in the entropy between $\mathcal{U}$ and $\mathcal{U}' = \mathcal{U} - \Delta\mathcal{U}$ ($\mathcal{U}' > \mathcal{U}_s$) is attributable to the tail. As such:

$$S(\mathcal{U}) - S(\mathcal{U}') = S_{tail}(\mathcal{U}) - S_{tail}(\mathcal{U}') \qquad (40)$$

Then, to prove that the entropy S does not change is equivalent to prove that the entropy in the tail does not change. To simplify this demonstration, and without loss of generality, we will assume that the frequency of tail words is one ($A = 1$). Due to 2), and the fact that new words are being added only to the tail, we can re-write the tail entropy difference as:

$$\begin{aligned} S_{tail}&(\mathcal{U}) - S_{tail}(\mathcal{U}') \\ &= \sum\nolimits_{i\in\mathcal{T}(\mathcal{U}')} p_i \cdot \log_2 p_i - \sum\nolimits_{i\in\mathcal{T}(\mathcal{U})} p_i \cdot \log_2 p_i \\ &= |\mathcal{T}(\mathcal{U}')| \cdot \frac{1}{O'} \log_2 \frac{1}{O'} - |\mathcal{T}(\mathcal{U})| \cdot \frac{1}{O} \log_2 \frac{1}{O} \\ &= |\mathcal{T}(\mathcal{U})| \cdot \left(\frac{\log_2 O'^{-1}}{O'} \cdot \frac{|\mathcal{T}(\mathcal{U}')|}{|\mathcal{T}(U)|} - \frac{\log_2 O^{-1}}{O} \right) \end{aligned}$$

where O' and O are the overall number of observations in the dictionaries $\mathcal{D}(\mathcal{U})$ and $\mathcal{D}(\mathcal{U}')$ respectively.

But since $|\mathcal{T}(U)| = O \cdot P_{tail}$, the previous expression is equivalent to:

$$\begin{aligned} O \cdot P_{tail} &\cdot \left(\frac{\log_2 O'^{-1}}{O'} \cdot \frac{O' P_{tail}}{O P_{tail}} - \frac{\log_2 O^{-1}}{O} \right) \\ &= O \cdot P_{tail} \cdot \left(\frac{\log_2 O'^{-1}}{O} - \frac{\log_2 O^{-1}}{O} \right) \\ &= P_{tail} \cdot \left(\log_2 O'^{-1} - \log_2 O^{-1} \right) \\ &= P_{tail} \cdot \log_2 \frac{O}{O'} \end{aligned} \qquad (41)$$

For a number sufficiently large of URIs (i.e $U \gg \Delta U$) the ratio $O/O' = O/(O - \Delta O) \rightarrow 1$, as and as such:

$$S_{tail}(\mathcal{U}) - S_{tail}(\mathcal{U} - \Delta\mathcal{U}) \simeq P_{tail} \times \log_2(1)) = 0 \qquad (42)$$

■

REFERENCES

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, Jul. 2009.

[2] H. Hindy, D. Brosset, E. Bayne, A. Seeam, C. Tachtatzis, R. Atkinson, and X. Bellekens, "A taxonomy and survey of intrusion detection system design techniques, network threats and datasets," 2018, *arXiv:1806.03517*. [Online]. Available: http://arxiv.org/abs/1806.03517

[3] N. Moustafa, J. Hu, and J. Slay, "A holistic review of network anomaly detection systems: A comprehensive survey," *J. Netw. Comput. Appl.*, vol. 128, pp. 33–55, Feb. 2019.

[4] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. IEEE Symp. Secur. Privacy*, 2010, pp. 305–316.

[5] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Comput. Secur.*, vol. 28, nos. 1–2, pp. 18–28, Feb. 2009.

[6] E. K. Viegas, A. O. Santin, and L. S. Oliveira, "Toward a reliable anomaly-based intrusion detection in real-world environments," *Comput. Netw.*, vol. 127, pp. 200–216, Nov. 2017.

[7] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy (ICISSP)*, 2018, pp. 108–116.

[8] G. F. Cretu, A. Stavrou, M. E. Locasto, S. J. Stolfo, and A. D. Keromytis, ''Casting out demons: Sanitizing training data for anomaly sensors,'' in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2008, pp. 81–95.

[9] J. Tolle, M. Jahnke, N. Felde, and P. Martini, ''Impact of sanitized message flows in a cooperative intrusion warning system,'' in *Proc. MILCOM*, Oct. 2006, pp. 1–7.

[10] C. Wressnegger, G. Schwenk, D. Arp, and K. Rieck, ''A close look onn-grams in intrusion detection: Anomaly detection vs. classification,'' in *Proc. ACM Workshop Artif. Intell. Secur. (AISec)*, 2013, pp. 67–76.

[11] P. Velarde-Alvarado, C. Vargas-Rosales, R. Martinez-Pelaez, H. Toral-Cruz, and A. F. Martinez-Herrera, ''An unsupervised approach for traffic trace sanitization based on the entropy spaces,'' *Telecommun. Syst.*, vol. 61, no. 3, pp. 609–626, Mar. 2015.

[12] M. Bermúdez-Edo, R. Salazar-Hernández, J. Díaz-Verdejo, and P. García-Teodoro, ''Proposals on assessment environments for anomaly-based network intrusion detection systems,'' in *Critical Information Infrastructures Security*, J. Lopez, Ed. Berlin, Germany: Springer, 2006, pp. 210–221.

[13] H. Holm, ''Signature based intrusion detection for zero-day attacks: (Not) a closed chapter?'' in *Proc. 47th Hawaii Int. Conf. Syst. Sci.*, Jan. 2014, pp. 4895–4904.

[14] R. Zuech, T. M. Khoshgoftaar, N. Seliya, M. M. Najafabadi, and C. Kemp, ''A new intrusion detection benchmarking system,'' in *Proc. 28th Int. Florida Artif. Intell. Res. Soc. Conf. (FLAIRS)*, 2015, pp. 252–255.

[15] R. A. A. Habeeb, F. Nasaruddin, A. Gani, I. A. T. Hashem, E. Ahmed, and M. Imran, ''Real-time big data processing for anomaly detection: A survey,'' *Int. J. Inf. Manage.*, vol. 45, pp. 289–307, Apr. 2019.

[16] C.-T. Huang and J. Janies, ''An adaptive approach to granular real-time anomaly detection,'' *EURASIP J. Adv. Signal Process.*, vol. 2009, no. 1, p. 7, Feb. 2009.

[17] K. Kishimoto, H. Yamaki, and H. Takakura, ''Improving performance of anomaly-based IDS by combining multiple classifiers,'' in *Proc. IEEE/IPSJ Int. Symp. Appl. Internet*, Jul. 2011, pp. 366–371.

[18] M. Tavallaee, N. Stakhanova, and A. A. Ghorbani, ''Toward credible evaluation of anomaly-based intrusion-detection methods,'' *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 40, no. 5, pp. 516–524, Sep. 2010.

[19] (1995). *Sask Dataset*. Accessed: Aug. 14, 2019. [Online]. Available: http://ita.ee.lbl.gov/html/contrib/Sask-HTTP.html

[20] A. Gharib, I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, ''An evaluation framework for intrusion detection dataset,'' in *Proc. Int. Conf. Inf. Sci. Secur. (ICISS)*, Dec. 2016, pp. 1–6.

[21] B. Du, Z. Wang, L. Zhang, L. Zhang, W. Liu, J. Shen, and D. Tao, ''Exploring representativeness and informativeness for active learning,'' *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 14–26, Jan. 2017.

[22] C. Guo, Y.-J. Zhou, Y. Ping, S.-S. Luo, Y.-P. Lai, and Z.-K. Zhang, ''Efficient intrusion detection using representative instances,'' *Comput. Secur.*, vol. 39, pp. 255–267, Nov. 2013.

[23] B. Sangster, T. O'Connor, T. Cook, R. Fanelli, E. Dean, C. Morrell, and G. J. Conti, ''Toward instrumenting network warfare competitions to generate labeled datasets,'' in *Proc. CSET*, Aug. 2009, pp. 1–6.

[24] H. G. Kayacik, A. N. Zincir-Heywood, and M. I. Heywood, ''On dataset biases in a learning system with minimum *a priori* information for intrusion detection,'' in *Proc. 2nd Annu. Conf. Commun. Netw. Services Res.*, 2004, pp. 181–189.

[25] R. A. Maxion and K. M. Tan, ''Benchmarking anomaly-based detection systems,'' in *Proc. Int. Conf. Dependable Syst. Netw. (DSN)*, 2000, pp. 623–630.

[26] K. M. Tan, K. S. Killourhy, and R. A. Maxion, ''Undermining an anomaly-based intrusion detection system using common exploits,'' in *Proc. Int. Workshop Recent Adv. Intrusion Detection*, 2002, pp. 54–73.

[27] W. Lee and D. Xiang, ''Information-theoretic measures for anomaly detection,'' in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2000, pp. 130–143.

[28] P. Tsangaratos and I. Ilia, ''Comparison of a logistic regression and naïve bayes classifier in landslide susceptibility assessments: The influence of models complexity and training dataset size,'' *Catena*, vol. 145, pp. 164–179, Oct. 2016.

[29] S. Mukherjee and N. Sharma, ''Intrusion detection using naive Bayes classifier with feature reduction,'' *Procedia Technol.*, vol. 4, pp. 119–128, Jan. 2012.

[30] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. Cambridge, MA, USA: MIT Press, 2009.

[31] Y. Dong, Y. Zhang, H. Ma, Q. Wu, Q. Liu, K. Wang, and W. Wang, ''An adaptive system for detecting malicious queries in Web attacks,'' *Sci. China Inf. Sci.*, vol. 61, no. 3, Feb. 2018, Art. no. 032114.

[32] H. Raza, G. Prasad, and Y. Li, ''EWMA model based shift-detection methods for detecting covariate shifts in non-stationary environments,'' *Pattern Recognit.*, vol. 48, no. 3, pp. 659–669, Mar. 2015.

[33] D. Hadžiosmanović, L. Simionato, D. Bolzoni, E. Zambon, and S. Etalle, ''N-gram against the machine: On the feasibility of the n-gram network analysis for binary protocols,'' in *Proc. Int. Workshop Recent Adv. Intrusion Detection*, 2012, pp. 354–373.

[34] R. Salazar-Hernández and J. E. Díaz-Verdejo, ''Hybrid detection of application layer attacks using Markov models for normality and attacks,'' in *Information and Communications Security*, M. Soriano, S. Qing, and J. López, Eds. Berlin, Germany: Springer, 2010, pp. 416–429.

[35] J. M. Estevez-Tapiador, P. Garcia-Teodoro, and J. E. Diaz-Verdejo, ''Detection of Web-based attacks through Markovian protocol parsing,'' in *Proc. 10th IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2005, pp. 457–462.

[36] T. Berners-Lee, R. Fielding, and L. Masinter, *Uniform Resource Identifier (URI): Generic Syntax*, document RFC 3986, IETF, 2005.

[37] *Faizaan24 Repository*. Accessed: Sep. 30, 2019. [Online]. Available: https://github.com/faizann24/

[38] *Attack Dataset*. Accessed: Sep. 30, 2019. [Online]. Available: https://hdvirtual.us.es/discovirt/index.php/s/xemaMdo8opraPgo

[39] M. Bekkar, H. K. Djemaa, and T. A. Alitouche, ''Evaluation measures for models assessment over imbalanced data sets,'' *J. Inf. Eng. Appl.*, vol. 3, no. 10, pp. 27–38, 2013.

[40] J. M. Lucas and M. S. Saccucci, ''Exponentially weighted moving average control schemes: Properties and enhancements,'' *Technometrics*, vol. 32, no. 1, pp. 1–12, Feb. 1990.

[41] C. C. Holt, ''Forecasting seasonals and trends by exponentially weighted moving averages,'' *Int. J. Forecasting*, vol. 20, no. 1, pp. 5–10, Jan. 2004.

[42] L. A. Jones, C. W. Champ, and S. E. Rigdon, ''The performance of exponentially weighted moving average charts with estimated parameters,'' *Technometrics*, vol. 43, no. 2, pp. 156–167, May 2001.

RAFAEL ESTEPA received the M.S. and Ph.D. degrees in telecommunication engineering from the University of Seville, in 1998 and 2002, respectively. In the past, he was working for two years as a Product Engineer with Alcatel, Spain. He has also been a Visitor with the Department of Applied Mathematics, Instituto Superior Tecnico (IST), Lisbon, and the Dublin Institute of Technology (DIT). He is currently an Associate Professor with the Department of Telematics Engineering, University of Seville. His research interests include the areas of networking, voice over IP (VoIP), the quality of service, wireless networks, unmanned aerial vehicles (UAVs), and cybersecurity.

JESÚS E. DÍAZ-VERDEJO received the M.S. and Ph.D. degrees in physics from the University of Granada, in 1989 and 1995, respectively. He is currently a Professor with the Department of Signal Theory, Telematics, and Communications, University of Granada. His initial research interest was focused on speech technologies. His current research and teaching interests are in the areas of networking and cybersecurity, especially in intrusion detection systems, network security monitoring, and traffic engineering.

ANTONIO ESTEPA received the M.S. and Ph.D. degrees in telecommunication engineering from the University of Seville, in 1998 and 2004, respectively. From 1998 to 2000, he was a software and network engineer with a software development company. In 2004, he was also a Visitor with the Department of Electrical Engineering and Computer Science, University of Minnesota, USA. He is currently an Associate Professor with the Department of Telematics Engineering, University of Seville. He has authored or coauthored in several conferences or journal articles. His research interests include the areas of telecommunication networks, with a particular emphasis on networking protocols, wireless networks, and cybersecurity.

GERMAN MADINABEITIA received the M.S. and Ph.D. degrees in telecommunication engineering from the Universidad Politecnica de Madrid, in 1986 and 2004, respectively. In the past, he was working for ten years as a product engineer in the industry. He is currently an Assistant Professor with the Department of Telematics Engineering, University of Seville. His research interests include the areas of networking, the Internet of Things, cybersecurity, and traffic engineering.